

CLAIMS

WHAT IS CLAIMED IS:

- Sub A1
- 1 1. A method of searching for a string in a lexical cache, comprising the computer-
2 implemented steps of:
3 generating a key based on the string; ✓
4 identifying a lexical container from among a plurality of lexical containers based on a
5 length of the key; and ✕
6 searching the lexical container for an entry associated with the string. ✓✓
 - Sub A2
 - 2 2. The method of claim 1, wherein the step of generating a key based on the string
includes the step of compressing the string to produce the key.
 - 1 3. The method of claim 2, wherein the step of compressing the string to produce the
2 key includes the step of performing an n-gram compression on the string.
 - 1 4. The method of claim 1, wherein the step of generating a key based on the string
2 includes the step of using the string as the key.
 - 1 5. The method of claim 1, wherein the step of identifying a lexical container includes
2 the steps of:
3 generating a prefix based on the key;
4 identifying the lexical container from among the plurality of the lexical containers
5 based on the length of the key and the prefix.

1 6. The method of claim 1, wherein:

2 the step of identifying a lexical container based on a length of the key includes the
3 step of identifying a hash table based on the length of the key, said hash table
4 containing sequences of slots for holding entries associated with strings, each of
5 said sequences of slots corresponding to a respective hash value; and
6 the step of searching the lexical container for an entry associated with said string
7 includes the steps of:
8 computing a hash value based on the key; and
9 searching the hash table based on the hash value for a slot holding an entry
10 associated with said string.

1 7. The method of claim 6, wherein the step of computing a hash value based on the
2 key includes the step of computing the hash value based on the key and a prime
3 number associated with the hash table.

1 8. The method of claim 7, wherein the step of searching the hash table based on the
2 hash value includes the steps of:
3 indexing one or more fixed regions of the hash table, each of the fixed regions having
4 the prime number of slots, based on the hash value to identify one or more
5 respective slots; and
6 inspecting the one or more respective slots for a respective key value matching the
7 key.

1 9. The method of claim 8, wherein the step of searching the hash table further
2 includes the step of searching for the key in a linked list of slots stored in an expansion
3 region of the hash table, if the key was not found in the one or more respective slots for
4 the key.

1 10. The method of claim 6, further including the step of, if an entry for the string is
2 not found at a first slot that corresponds to the hash value, but is found in a slot that
3 belongs to a sequence of slots that correspond to keys that produce said hash value, then
4 moving a relative position of the entry for the string within the sequence of slots toward
5 the beginning of the sequence of slots.

1 11. The method of claim 6, further comprising the step of initializing a descriptor for
2 the hash table, said descriptor storing a reference to the hash table and parameters for the
3 hash table;

4 wherein the step of identifying a hash table includes the step of identifying a
5 descriptor indicating the hash table and a prime number.

1 12. The method of claim 11, wherein the step of initializing a descriptor for the hash
2 table includes the step of initializing a prime number for use in computing a hash value.

1 13. The method of claim 11, wherein the step of initializing a descriptor for the hash
2 table includes the step of initializing a maximum number of slots for the hash table.

1 14. The method of claim 11, wherein the step of initializing a descriptor for the hash
2 table includes the step of initializing a maximum length of the sequences of slots for the
3 hash table.

15. A method of searching for a string in a lexical cache, comprising the computer-implemented steps of:

compressing the string to generate a key; ✓

identifying a hash table from among a plurality of hash tables based on a length of the

key, said hash table containing sequences of slots for holding respective key values, each of said sequences of slots corresponding to a respective hash value;

computing a hash value based on the key;

using said hash value to locate a beginning of the particular sequence of slots that correspond to said hash value;

searching the particular sequence of slots for a slot holding a key value matching the key; and ✓

if a slot having a key value matching the key is found in the particular sequence of slots, but is not at the beginning of said particular sequence of slots, then moving a relative position of the key value within the particular sequence of slots toward the beginning of the particular sequence of slots.

16. A computer-readable medium bearing instructions for searching for a string in a lexical cache, said instructions arranged, when executed by one or more processors, to cause the one or more processors to perform the steps of:

generating a key based on the string;

identifying a lexical container from among a plurality of lexical containers based on a length of the key; and

searching the lexical container for an entry associated with the string.

17. The computer-readable medium of claim 16, wherein the step of generating a key based on the string includes the step of compressing the string to produce the key.

1 18. The computer-readable medium of claim 17, wherein the step of compressing the
2 string to produce the key includes the step of performing an n-gram compression on the
3 string.

1 19. The computer-readable medium of claim 16, wherein the step of generating a key
2 based on the string includes the step of using the string as the key.

1 20. The computer-readable medium of claim 16, wherein the step of identifying a
2 lexical container includes the steps of:
3 generating a prefix based on the key;
4 identifying the lexical container from among the plurality of the lexical containers
5 based on the length of the key and the prefix.

1 21. The computer-readable medium of claim 16, wherein:
2 the step of identifying a lexical container based on a length of the key includes the
3 step of identifying a hash table based on the length of the key, said hash table
4 containing sequences of slots for holding entries associated with strings, each of
5 said sequences of slots corresponding to a respective hash value; and
6 the step of searching the lexical container for an entry associated with said string
7 includes the steps of:
8 computing a hash value based on the key; and
9 searching the hash table based on the hash value for a slot holding an entry
10 associated with said string.

1 22. The computer-readable medium of claim 21, wherein the step of computing a
2 hash value based on the key includes the step of computing the hash value based on the
3 key and a prime number associated with the hash table.

1 23. The computer-readable medium of claim 22, wherein the step of searching the
2 hash table based on the hash value includes the steps of:
3 indexing one or more fixed regions of the hash table, each of the fixed regions having
4 the prime number of slots, based on the hash value to identify one or more
5 respective slots; and
6 inspecting the one or more respective slots for a respective key value matching the
7 key.

1 24. The computer-readable medium of claim 23, wherein the step of searching the
2 hash table further includes the step of searching for the key in a linked list of slots stored
3 in an expansion region of the hash table, if the key was not found in the one or more
4 respective slots for the key.

1 25. The computer-readable medium of claim 21, wherein said instructions are further
2 arranged to cause the one or more processors to perform the step of, if an entry for the
3 string is not found at a first slot that corresponds to the hash value, but is found in a slot
4 that belongs to a sequence of slots that correspond to keys that produce said hash value,
5 then moving a relative position of the entry for the string within the sequence of slots
6 toward the beginning of the sequence of slots.

1 26. The computer-readable medium of claim 21, wherein said instructions are further
2 arranged to cause the one or more processors to perform the step of initializing a
3 descriptor for the hash table, said descriptor storing a reference to the hash table and
4 parameters for the hash table;
5 wherein the step of identifying a hash table includes the step of identifying a
6 descriptor indicating the hash table and a prime number.

1 27. The computer-readable medium of claim 26, wherein the step of initializing a
2 descriptor for the hash table includes the step of initializing a prime number for use in
3 computing a hash value.

1 28. The computer-readable medium of claim 26, wherein the step of initializing a
2 descriptor for the hash table includes the step of initializing a maximum number of slots
3 for the hash table.

1 29. The computer-readable medium of claim 26, wherein the step of initializing a
2 descriptor for the hash table includes the step of initializing a maximum length of the
3 sequences of slots for the hash table.

1 30. A computer-readable medium bearing instructions for searching for a string in a
2 lexical cache, said instructions arranged, when executed by one or more processors, to
3 cause the one or more processors to perform the steps of:
4 compressing the string to generate a key;
5 identifying a hash table from among a plurality of hash tables based on a length of the
6 key, said hash table containing sequences of slots for holding respective key
7 values, each of said sequences of slots corresponding to a respective hash value;
8 computing a hash value based on the key;
9 using said hash value to locate a beginning of the particular sequence of slots that
10 correspond to said hash value;
11 searching the particular sequence of slots for a slot holding a key value matching the
12 key; and
13 if a slot having a key value matching the key is found in the particular sequence of
14 slots, but is not at the beginning of said particular sequence of slots, then moving

- 15 a relative position of the key value within the particular sequence of slots toward
16 the beginning of the particular sequence of slots.

Addr 15
addr 02

Addr
017

00000000000000000000000000000000